

Mobile context-based framework for threat monitoring in urban environment with social threat monitor

Szymon Bobek · Grzegorz J. Nalepa · Antoni Ligęza ·
Weronika T. Adrian · Krzysztof Kaczor

Received: 28 December 2013 / Revised: 24 May 2014 / Accepted: 27 May 2014 /
Published online: 29 May 2014
© Springer Science+Business Media New York 2014

Abstract Engaging users in threat reporting is important in order to improve threat monitoring in urban environments. Today, mobile applications are mostly used to provide basic reporting interfaces. With a rapid evolution of mobile devices, the idea of context awareness has gained a remarkable popularity in recent years. Modern smartphones and tablets are equipped with a variety of sensors including accelerometers, gyroscopes, pressure gauges, light and GPS sensors. Additionally, the devices become computationally powerful which allows for real-time processing of data gathered by their sensors. Universal access to the Internet via WiFi hot-spots and GSM network makes mobile devices perfect platforms for ubiquitous computing. Although there exist numerous frameworks for context-aware systems, they are usually dedicated to static, centralized, client-server architectures. There is still space for research in the field of context modeling and reasoning for mobile devices. In this paper, we propose a lightweight context-aware framework for mobile devices that uses data gathered by mobile device sensors and performs on-line reasoning about possible threats based on the information provided by the Social Threat Monitor system developed in the INDECT project.

Keywords Context-awareness · Mobile computing · GIS · Knowledge management · INDECT

The research presented in this paper is carried out within the EU FP7 INDECT Project: “Intelligent information system supporting observation, searching and detection for security of citizens in urban environment” (<http://indect-project.eu>). The paper is supported by the AGH UST Grant 11.11.120.859.

S. Bobek (✉) · G. J. Nalepa · A. Ligęza · W. T. Adrian · K. Kaczor
AGH University of Science and Technology, al. Mickiewicza 30, 30-059, Krakow, Poland
e-mail: szymon.bobek@agh.edu.pl

A. Ligęza
e-mail: gjn@agh.edu.pl

W. T. Adrian
e-mail: wta@agh.edu.pl

K. Kaczor
e-mail: kk@agh.edu.pl

1 Introduction

Distributed reporting and notification systems for citizen security became common and widely expected and adopted in recent years. Within the scope of the INDECT¹ project such solutions are being developed and evaluated. One of the principal objectives of the project include engaging citizens into active participation in the authorities efforts to provide instant notification for a number of security threats in a given neighborhood or a wider location. The threats can be considered in various categories, such as crime, natural disasters, accidents or traffic related events.

A system called *Social Threat Monitor* (STM) that meets the above mentioned needs was developed [24]. STM is a GIS-based solution that assists citizens in reporting security threats together with their severity and location. The threats are classified using a general top-level ontology, with domain ontologies supporting the detailed specification of threats. The information about the threats is stored in a knowledge base of the system which allows for lightweight reasoning with the gathered facts. All the threats can be located on a web-accessible map that can be analyzed by a group of users, e.g., police officials, regular citizens, etc.

The current version of the system is a web-based solution, composed of a server-side GIS-based service providing access to the knowledge base and a web client. Therefore, a standard-compliant web browser is expected to be used as the main user interface. Another method for interfacing with the system on the application level is provided by a dedicated API that allows posing queries and making updates to the knowledge base.

An apparent limitation of the current system is related to the use of mobile devices on the client side [6]. In the first generation of the system, an implicit assumption that the user has a standard web browser available was made. Moreover, this browser should be used with a regular (for desktop and laptop computers) point-and-click user interface. However, currently the most common use case scenario includes the use of a mobile handheld device, such as a smartphone or a tablet, with a number of multimodal interfaces and sensors. Therefore, a need for a new front-end for the system became apparent. The principal objective of this paper is to propose and discuss a design of a prototype of such a system. It uses the context-aware application paradigm that improves usability from the user perspective, and simplifies the use of multi sensor data available on the mobile devices. This paper is partially based on the paper previously presented at the MCSS 2013 conference [6]. However, it was largely reworked and extended with further research including the implementation of a prototype of the discussed framework. Furthermore, identification of requirements needed for deployment on mobile platforms and practical evaluation on selected devices was performed.

The rest of the paper is organized as follows. In Section 2, related works including the issues of context modeling and reasoning in mobile devices are discussed. In Section 3 the Social Threat Monitor system is described, and its weaknesses with respect to mobile environments are presented. This leads to presenting the motivation of the development of a new STM front-end in Section 4. The architecture of the system is presented in Section 5. The core element of an architecture which is the inference service was presented and discussed in Section 9. A practical use case scenario is briefly introduced in Section 10. The evaluation of the framework on a prototype of a context-aware mobile application is presented in Section 11. Finally, summary and directions for future work are given in Section 12.

¹ See <http://indect-project.eu>.

2 Related works

The notion of context has been important in conceptualization of systems for many years. Research in the area of pervasive computing and ambient intelligence aims to make use of context information to allow devices or applications behave in a context-aware way. Dey [12] defines context as *any information that can be used to characterize the situation of an entity*, where *an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*.

The *information* in Dey's definition may be:

- location of the user (spatial context),
- presence or absence of other devices and users (social context),
- time (temporal context),
- user *behavior* or *activity* (activity recognition, behavior modeling), and
- other environmental data gathered by microphones, light sensors, etc.

Raw information captured by the device sensors is usually useless without further preprocessing and interpretation. Thus, the main challenges in context-aware systems are context modeling and context-based reasoning. There is a lot of research regarding context modeling. Various methods of knowledge representation were used, e.g., rules and logic [13, 27, 39], ontologies [10, 42], object-oriented languages (CML, ORM) [19], context lattices [46] or processes [21].

In the area of context-based reasoning, the following approaches were developed: machine learning and probabilistic inference [8, 45], decision trees [26], rule-based and logic-based reasoning [33]. Although there are many frameworks and middlewares developed for context-aware systems, they are usually limited to a specific domain and designed without taking into consideration mobile platforms. Examples include CoBrA [9] and SOUPA [10] for building smart meeting rooms, GAIA [40] for active spaces, and Context Toolkit [13].

In recent years, a lot of development was devoted to building applications that use mobile devices to monitor and analyze various user contexts. The availability of application distribution platforms for mobile operating systems, e.g. Google Play for Android stimulated the adoption of such solutions. However, most of them focus only on a very narrow application area of context awareness. The SocialCircuits platform [11] uses mobile phones to measure social ties between individuals, and uses long- and short-term surveys to measure the shifts in individual habits, opinions, health, and friendships influenced by these ties. Sociometric badge [38] was designed to identify human activity patterns, analyze conversational prosody features and wirelessly communicate with radio base-stations and mobile phones. Sensor data from the badges was used in various organizational contexts to automatically predict employee's self-assessment of job satisfaction and quality of interactions. Eagle and Pentland [16] used mobile phone's Bluetooth transceivers, phone communication logs and cellular tower identifiers to identify the social network structure, recognize social patterns in daily user activity. Besides research projects, there exist also a variety of application that are used for gathering information about context from mobile devices, like SDCF [4], AWARE², JCAF [5], SCOUT [44], ContextDriod [43], Gimbal³. These are

²See: <http://www.awareframework.com>.

³See: <http://www.gimbal.com>.

mostly concerned with low-level context data acquisition from sensors, suitable for further context identification. On the other hand, they do not provide support nor any methodology for creating complex and fully customizable context-aware systems and do not provide any mechanisms for limiting energy consumption of the system. So, there is still space for research in a field of lightweight context modeling and context reasoning targeted at mobile devices. Some attempts were made to develop such frameworks, like SOCAM [18], or Context Torrent [20]. However, these frameworks do not provide full support for all of the challenges that we believe are crucial for mobile computing, with respect to the context modeling and context-based reasoning:

1. **energy efficiency** – most of the sensors, when turned on all the time, decrease the mobile device battery level very fast. This reflects on usability of the system and ecological aspects regarding energy saving.
2. **privacy** – most of the users do not want to send information about their location, activities, and other private data to external servers. Hence, the context reasoning should be performed by the mobile device.
3. **resource limitations** – although mobile phones and tablets are becoming computationally powerful, the context aware system has to consume as low CPU and memory resources as possible in order to be transparent to the user and other applications.

All of these require from the modeling language and inference engine to be simple and lightweight. Aforementioned challenges were usually approached by the programmers at the very last phase of the development of context-aware application, or were not approached at all. We believe that solutions to these challenges should be provided by the framework architecture. This will oblige the programmer to build context-aware application in an efficient way, making the development easier and less error prone.

3 Social threat monitor

Social Threat Monitor is a semantically enriched system for collaborative knowledge management. It was developed to allow local communities to share information about road traffic dangers and threats, i.e. closed roads, holes in the pavements and streets, dangerous districts or events that impede normal traffic. STM is aimed to be a community portal that allows citizens to participate and cooperate in order to improve the security in the urban environment.

Several prototypes of the system were developed [2, 3, 41]. Different versions of the STM used diversified intelligent processing to provide possibly most useful knowledge to the users. Categorization of threats and possibility of inferring new facts based on the ones entered by users was introduced. In [3], the whole system is based on a threat ontology, and the information returned to users are inferred by a reasoner.

In a recent prototype [23], social aspects of the system were emphasized. The Web-based user interface was simplified and a large map has become the main part of it (see Fig. 1). Users are able to browse threats in selected areas and submit information about a threat by placing it directly on the map. Adding comments and photos, discussions and votes aim to strengthen the credibility of the system and encourage users to contribute.

Public services may also inform people about dangers, monitor threats and decide if an intervention is needed. The system provides a mechanism to define *roles* to users and assign them to *groups*. This allows police to use the STM for their investigations, ensuring that

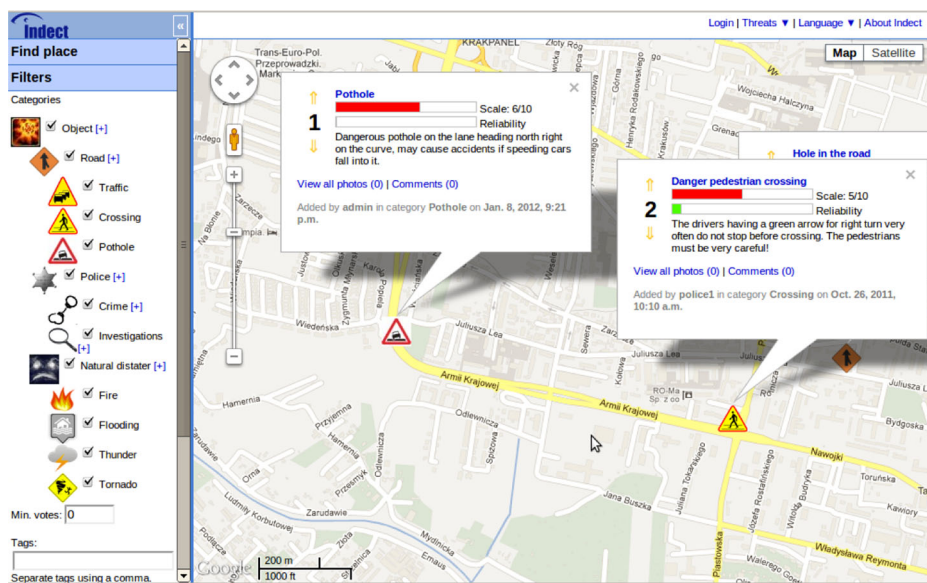


Fig. 1 User Interface of the Social Threat Monitor [6]

some landmarks are only visible to their group and not to the regular users. Role mechanism may be viewed as a first step towards contextualizing the system.

Besides groups that can be defined by a local administrator, by default the system provides the following groups of users:

- *Guest* is a user with the anonymous web account. He or she is able to use basic features of the application. In order to gain more privileges, a guest need to register and log in into the system.
- *Member* is a user with registered account in the system. With this account user can add threats, manage his own threats, comment and vote threats of other users and edit his profile.
- *Services User* is a special account with features helping threats monitoring.
- *Moderator* is a user with full access to threat records, able to ban users.
- *Administrator* is a user with full access to the application.

In order to ensure a better exchange of information between the STM and other applications, an HTTP-based API has been developed [2]. Parts of the system data can be imported and exported to JSON, XML and YAML formats. Standardized knowledge representation that uses *attribute-value* pairs describing threats allows for using the system knowledge in various semantic applications where triples of the form: *object-attribute-value* or *subject-predicate-object* are used. External systems can communicate with STM and use it as a web service. Part of the functionality of the STM is available for external applications without authorization. For instance, by preparing an appropriate request conforming to the system API, one can get all the threats defined for a given location or filter. This information may be further processed according to the application needs. In order to use the whole functionality, including adding threats to the system, the application must be authorized.

The API uses POST requests and HTTP cookies. All responses are in JSON format. Each successful request returns HTTP 200 header and the 400 header is returned, if a method does not exist. Moreover, if an anonymous user wants to access a method that requires authorization, a HTTP 200 header with appropriate content is returned. The system API provides three basic methods: 1) logging in, 2) adding a new threat, and 3) retrieving existing threats. Each method accepts arguments that must be sent using POST or COOKIES.

Although the STM API provides sufficient functionality for threat reporting and monitoring, the user interface of the system requires a lot of user interaction to present desired results. This is a serious limitation especially for mobile systems, where navigation should be fast and as simple as possible. The following Section describes in details the motivation of the work presented in this paper.

4 Motivation

The main limitation of the Social Threat Monitor remained the user interface (UI) which should be adaptable to various hardware platforms including desktop and mobiles. In fact, so far the system was not designed to use mobile devices on the client side. Nevertheless, currently the most common use case scenario includes the use of a mobile handheld device, such as a smartphone or tablet. Such a device has a number of multimodal interfaces and sensors, e.g. gesture based interface, GPS, etc. Therefore, a need for a new front-end for STM became apparent.

The idea comes down to propose a prototype of such a front-end. It is based on the context-aware application paradigm that improves usability from the user perspective. Moreover, it simplifies the use of multi sensor data available on the mobile devices. The general concept is presented in Fig. 2.

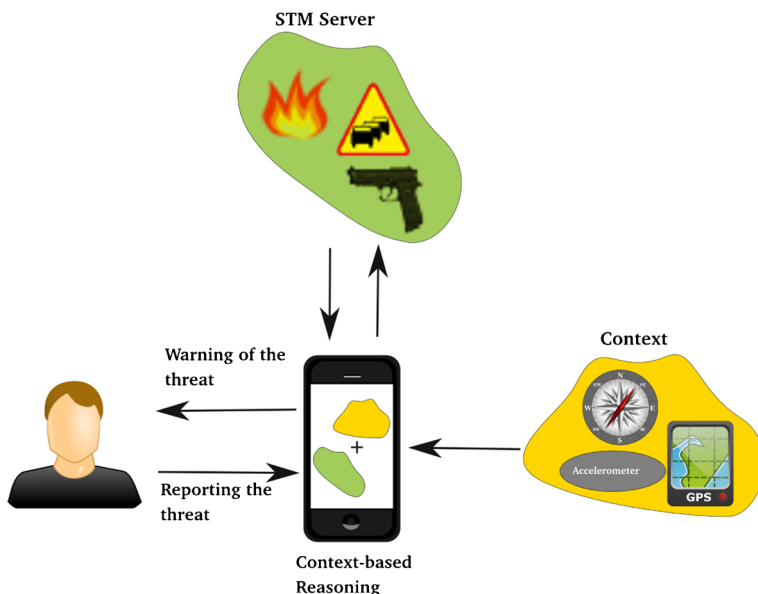


Fig. 2 Context-based front-end for Social Threat Monitor [6]

Being in a given situation and location users can be automatically notified by their mobile device about the threats relevant to them and the situation. Relevance to the person may be related to their role defined in the STM system, as well as the context, e.g. a person who walks should not be bothered by warnings relevant only to drivers in the same location. The use of data fusion from the sensors and multimodal interfaces of the mobile device allows to limit the amount of data a user would have to provide to the system. In fact, we propose a major paradigm shift on the front-end side. Whereas the original interface of STM was mostly query-based, here we propose a push-based UI where the user is automatically notified only about the information relevant to him or her. The system automatically uses the context data from the mobile device, as well as the data acquired from the STM server to perform reasoning for the user. In the following section, the architecture of the system is discussed.

5 Context-based STM front-end architecture

The proposed system is based on a service-oriented architecture (see Fig. 3). It consists of three main elements:

1. sensors service – responsible for gathering data from sensors and performing initial preprocessing of them,
2. inference service – responsible for context-based reasoning and knowledge management. It provides TCP/IP API for context-aware applications,
3. working memory middleware – acting as an intelligent proxy between sensors service and the inference service. It provides intelligent sensor management to minimize energy consumption and improve responsiveness by preparing required context in advance.

6 Sensors Service

The Sensors Service gathers data directly from mobile device sensors. Due to the different possible sensor types (GPS, Accelerometer, Bluetooth), different methods for interpreting this data are required. Hence, each sensor has its own interpreter module that is responsible for initial preprocessing of the raw data. Data preprocessing is triggered by the Working Memory Middleware.

7 Inference service

The inference service is responsible for performing reasoning, based on the model (knowledge base) and the working memory elements (facts). The service is capable of managing many models transparently switching between them. The reasoning task is performed by HeaRT [1]. It is a lightweight rule-based inference engine that uses XTT2 [37] notation for knowledge representation. It is written in Prolog and can be installed on mobile device together with tuProlog⁴ interpreter, providing autonomous inference service. Moreover, the

⁴See <http://alice.unibo.it/xwiki/bin/view/Main/>.

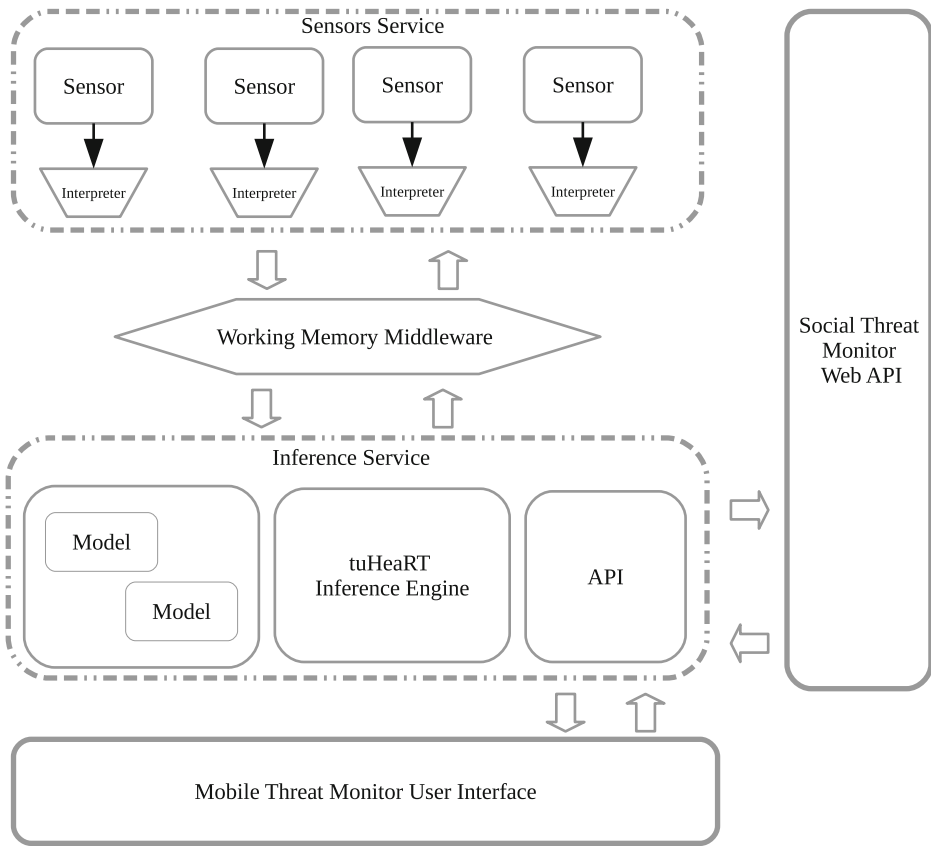


Fig. 3 Architecture of the mobile context aware framework for Social Threat Monitor

HeaRT inference engine, in contrary to other rule-based inference engines, provides custom verification module that can be used for automatic optimization of knowledge base (see Section [29] for details).

The inference service provides a TCP/IP interface for context-aware applications that may query HeaRT for important information. An exemplary query may concern listing all possible threats. This will require the inference engine to determine a context of the user (decide if a user is a driver, a cyclist, a pedestrian, decide where the user is, or where he or she will be in the nearest future). Based on these facts and on the data pulled from Social Threat Monitor system, the inference service will return the list of all the threats relevant for the user.

8 Working memory middleware

The Working Memory Middleware is responsible for exchanging information between sensors service and inference service. The working memory is shared between all models stored

within the inference service, acting as a *knowledge cache*. Therefore, it minimizes the number of required requests to the sensors service, improving power efficiency of the entire system.

The prototype of the Working Memory Middleware, that makes use of machine learning approach to discover sensor usage patterns in order to minimize energy usage, was described in details in [7]. The middleware uses modified linear regression algorithm that learns sensor usage patterns and adapt to it by minimizing queries to the sensor layer when it is less likely that the sensor will provide important data. It automatically generates a model of usage habits from historical data and based on that model, it adjusts the sampling rates for the sensors appropriately. It improves power efficiency of the system, since sampling rates are not fixed but learned from the usage patterns. On the other hand it may help in increasing responsiveness of the system, since the learned model allows predicting not only future sensor activity but also context-aware application needs. Hence, it is possible to get the desired context in advance, before the application actually requests it. It can be especially useful in cases when context cannot be obtained by the middleware directly from the sensor layer, but has to be for example downloaded over the Internet.

9 Context-based knowledge management

The context aware framework presented in this paper uses the XTT2 [37] notation for knowledge representation. It is a visual knowledge representation and design method for rule-based systems [25, 34] where rules are stored in tables connected with each other creating a graph (see Fig. 4).

The XTT2 has a textual representation called HMR. An example of a rule written in HMR language is presented below. The rule is referenced in Fig. 4, in table *Today*.

```
xrule Today/1: [day in [sat,sun]] ==>[today set weekend] .
```

The HMR representation is used by the HearT inference engine, which provides several inference modes, including:

Fixed-Order which is the simplest algorithm that consists of a hard-coded order of inference, in such way that every table is assigned an integer number; all the numbers are different from one another. The tables are fired in order from the lowest number to the highest one.

Data-Driven which can be used to find all possible information that can be inferred from given data. This mode is equivalent to forward chaining inference.

Goal-Driven which can be used to find only a specific information that can be inferred from a specific subset of XTT2 tables. This mode is equivalent to backward chaining inference.

Token-Driven which is based on monitoring the partial order of inference defined by the network structure with tokens assigned to tables. A table can be fired only when there is a token at each input. Intuitively, a token at the input is a kind of a flag signaling that the necessary data generated by the preceding table is ready for use.

These inference modes allow the efficient reasoning in structured knowledge bases, like XTT2. Only the tables that lead to desired solution are fired, and no rules are fired without a purpose, making the inference process less resource-consuming. Detailed description of the inference algorithms for XTT2 rule bases, can be found in [28]. For the purpose of context

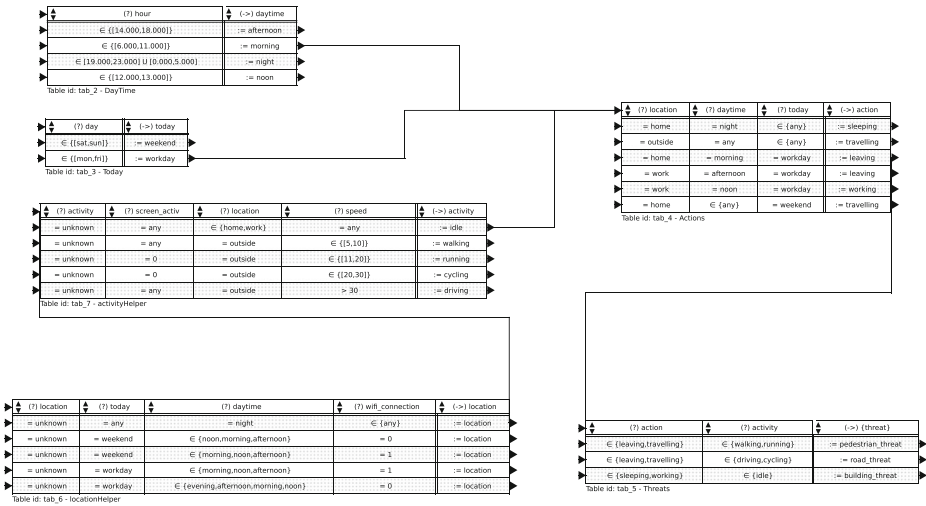


Fig. 4 Example of a model for a mobile threat monitor

processing a process-based description could also be used, see [22, 30]. In fact high level dependencies between context variables could be conceptualized on a high level [36].

The HeaRT inference engine provides a callback mechanism that allows to query external sources for information. The external source could be: database, user, or in our case working memory middleware and Social Threat Monitor system. Callbacks are associated with attributes, defined as e.g.:

```
xattr [ name: day,
        class: simple,
        type: day,
        comm: in,
        callback: [ask_working_memory, [day]]
      ] .
```

The *comm* element in the attribute definition determines behavior of a callback. There are three different types of such behavior:

- comm: in – the callback is obliged to pull the value of the attribute from the fact base,
- comm: out – the callback is obliged to push the value of the attribute to the fact base,
- comm: inter – the callback should not assert nor retract any information to the fact base.

More about callback mechanism can be found in [32].

9.1 HeaRT implementation for mobile platforms

HeaRT inference engine was originally designed and implemented as a desktop, standalone application. It was written in SWI-Prolog that allows for rapid prototyping of HeaRT plugins, and portability between different operating systems. HeaRT architecture assumes that it can be used either as a reasoning server that communicates with its clients over TCP/IP protocol, or can be embedded into application and communicate with it thanks to callback mechanism. Although this was perfectly understandable in static, desktop architecture,

mobile systems have different needs and assumptions. The following requirements and challenges were identified as crucial for inference service in mobile environments:

- (R1) Resource limitations. Reasoning should consume as least resources as possible to work transparently in the background.
- (R2) System responsiveness. The inference layer has to work under soft real-time constraints. Mobile environment is highly dynamic, and the inference layer should follow rapid changes of context in such an environment.
- (R3) Data privacy. The reasoning service should not send any confidential data to the external servers, but perform all the reasoning locally.
- (R4) Robustness. It should work properly when the contextual data is incomplete or uncertain. This requirement is beyond of the scope of this article and is just briefly described in the following paragraphs.
- (R5) Intelligibility. It should be able to explain its decision to the user, and thus improve intelligibility of the system.
- (R6) Portability. The inference engine should be portable and as independent of the operating system as possible.

These requirements were foundations for developing a mobile version of HeaRT that should be 1) lightweight, 2) work locally as a service for mobile context-aware applications 3) allow for fast responses to rapidly changing context. This however appeared to be a non trivial task, and currently the mobile version of HeaRT is still under development.

HeaRT is a lightweight rule-based inference engine written in Prolog that is able to provide autonomous inference service, independent on any external knowledge base. This guaranties that the context-based reasoning can be performed locally, and thus the requirement R3 is fulfilled.

What is more, rule-based reasoners like HeaRT have high capabilities of explaining their decisions. This allows almost immediate satisfaction of requirement R5, as the reasoning process performed by HeaRT can be retrieved and presented to the user in a form of explanation of the system decisions. This improves the intelligibility of the system, defined as an ability of the system to being understood [14]. An example of such explanatory mechanism was presented in Section 11 in Fig. 5.

The portability requirement (R6) is one of the most difficult to be satisfied in the area of mobile and embedded systems. Each of the existing operating systems for mobile devices use different native programming language. Thus, it is not possible to use one of the most popular programming languages like Java, Objective C or C#, as it will only be supported by fraction of mobile systems. This is why, we decided to use independent platform like Prolog, which can work as a virtual platform on the top of a native operating system. However, such a migration is not trivial and requires a lot of effort to allow the rest of the requirements defined in this section hold.

The same problems are connected with the requirements R1 and R2. Although they are easily satisfiable for HeaRT in a desktop environment, they no longer hold on mobile platform. The early attempt to migration from desktop to mobile platforms is briefly discussed in Section 11.

The requirement R4 is connected with the nature of the mobile environment. Although the mobile devices are currently equipped with variety of sensors, the contextual data provided by them is not always complete nor certain. While important, this requirement is beyond the scope of this article.

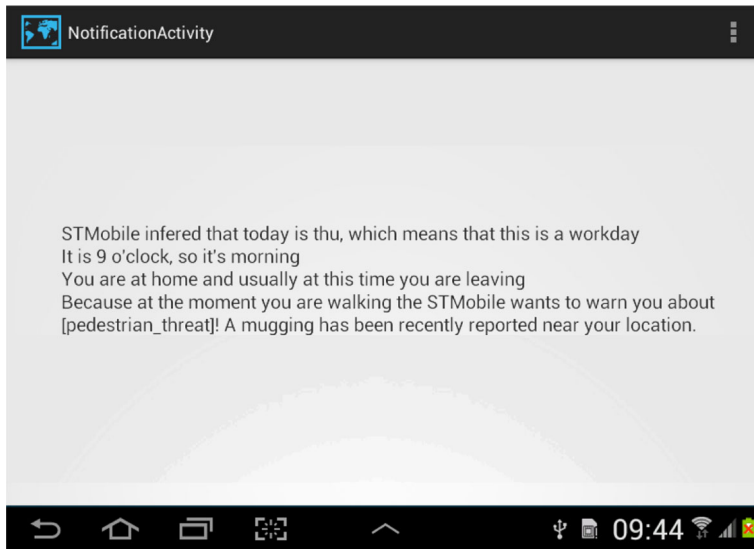


Fig. 5 Explanation of the inference engine decision built from the inference trace

10 Use case scenario

An exemplary XTT2 model (see Fig. 4) presented in this section allows to alert users about threats in a context-aware way. The system takes into consideration spatial (localization of the user) and temporal (time of a day) contexts, as well as user activities. This allow the intelligent threats filtering. For instance, the model will prevent from warning a user who is driving a car about threats that are applicable to pedestrians only. This is achieved by selecting only these rules that are valid in the current context.

Information about threats is fetched from Social Threat Monitor system via callbacks using the STM API (see Section 3 for details). Information about user localization, time of a day, and user activities is pulled from a working memory middleware via callback mechanism. The working memory middleware obtains this information from sensors interpreters (for example: location from GPS sensor interpreter, activity like walking or running from accelerometer sensor interpreter, etc.).

Taking into consideration the example from Fig. 4, and assuming that it is Monday, 8 o'clock am, and the user is driving a car, the system will process the following rules: (1) rule 1 from *DayTime* table, (2) rule 2 from *Today* table, (3) rule 4 from *Actions* table, (4) and rule 2 from *Threats* table.

This inference chain will trigger several callbacks, including one that is assigned to *road_threats* in the *Threats* table. The callback will fetch all road threats from Social Threat Monitor system that are located near the user and assign it to *road_threats* attribute.

The model presented in Fig. 4 contains two additional tables called *activityHelper* and *locationHelper*. This tables will be processed in cases when location or activity provider is not available, or their predictions are very uncertain. In such case, a coarse location and activity can be inferred based on some a priori defined dependencies between sensors like: *when there is a WiFi connection and the user is in the middle of the work day it is very likely*

that he is at work. This allows for a very basic handling of incomplete or uncertain data (R4 from Section 9.1). However, in a long time perspective this is only a walk-around, and additional mechanisms dedicated for handling such anomalies have to be incorporated into the model in the future.

The application that implements Mobile Threat Monitor interface will be able to pull all information about threats from inference service via TCP/IP API and display it to the user. An example callback that fetches all road threats near a specified GPS location is presented below:

```
xcall road_threats_callback: [AttName] >>>
    (jpl_new('skeleton.RequestInterface', [], T),
    jpl_call(T,
        request,
        ['callbacks.input.IndectAPI', '[roadThreats, GPSLocation]',
        Answer),
    atom_to_term(Answer, Answer2, _),
    alsv_new_val(AttName, Answer2)).
```

The callback uses Java library for callbacks designed for HeaRT inference engine. The callback mechanism is not only a tool for fetching data from other system components. It can also provide a mediation support with user, and thus improve intelligibility and usability of the system [15].

11 Solution evaluation

The prototype system presented in this paper offers an important improvement over the basic STM system. The use of mobile devices as interfaces to the STM system improves the usability of the system. It makes it easier for users to actually input the information into the system. Moreover, the mobile interface allows users to receive notifications on time and considering the relevant context, e.g. location. For the evaluation a Samsung Galaxy Tab II (GT-P5100) was used, with Android 4.1.2 installed.

11.1 Inference service

The inference service was implemented with a HeaRT rule-based reasoner. As a Prolog interpreter for HeaRT inference engine, tuProlog 2.8.0 was used, as the official release, which at the time of writing is 2.7.2, has critical bug in retract predicate.

The XTT2 model presented in Fig. 4 was used to provide context-based threats filtering. The reasoning is triggered by the new contextual information delivered by the Aware framework [17]. The inference process is performed locally, to preserve the requirement R3. The threats are fetched from STM via STMJava API and are presented to the user in a form of system notification (See Fig. 5).

When a user decides to check details of the threat, a new activity is presented, where the full explanation of the decision made is presented to him. The explanation is made based on the trace of the inference process which is delivered to the application as a system state. An example of such trace is presented below.

```
10 ?- xstat current: Value.
Value = [hour, 9] ;
Value = [day, thu] ;
Value = [location, home] ;
```

```

Value = [activity, walking] ;
Value = [daytime, morning] ;
Value = [today, weekday] ;
Value = [action, leaving] ;
Value = [threat, [pedestrian, threat]] ; false.

```

The explanation based on the state trace from the listing above is presented in the Fig. 5. This preserves the requirement R5, providing basic intelligibility feature to the system.

Responsiveness of the reasoning service was calculated as an average of time required to deliver the value of the *threat* attribute after invoking the inference process. The average time for the inference for the model presented in Figure 4 equals 1800 milliseconds. The results was surprisingly bad comparing to the desktop environment. However, the low responsiveness was caused by the optimization issues in tuProlog implementation, not by the HeaRT reasoning engine. Nevertheless, this makes the requirement R1 not satisfied for the evaluation use case.

The CPU time required to deliver the answer by the reasoning engine was also very high. The CPU usage of the inference service is presented in Fig. 6. The issue of very high CPU load is again connected with not optimal tuProlog implementation, that takes a lot of time to execute the Prolog queries. Thus, the requirement R2 for the Inference service cannot hold. Requirements concerning robustness and portability of the system are beyond of the scope of this paper and are included in future work.

While the current prototype is still a work in progress, it has certain limitations. First of all it was tested with a limited number of uses cases (user context descriptions). Furthermore, the mobile devices used were mostly the Samsung Galaxy Tab II series. Finally, the representation of threats is limited and does not integrate the ontological description developed for the STM [3].

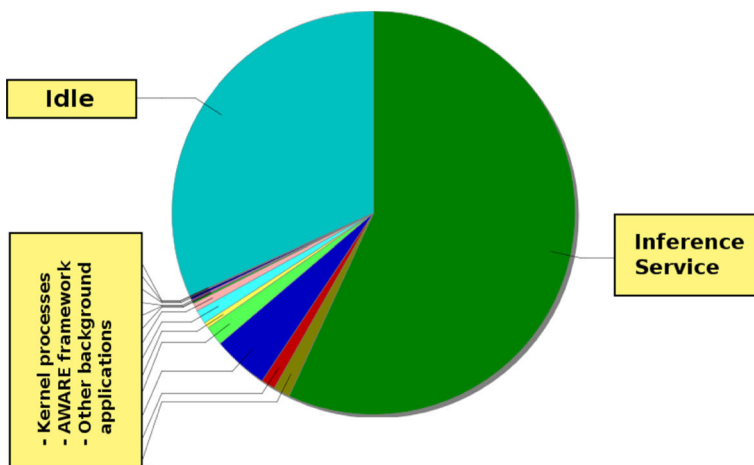


Fig. 6 CPU load for the inference engine working as a service

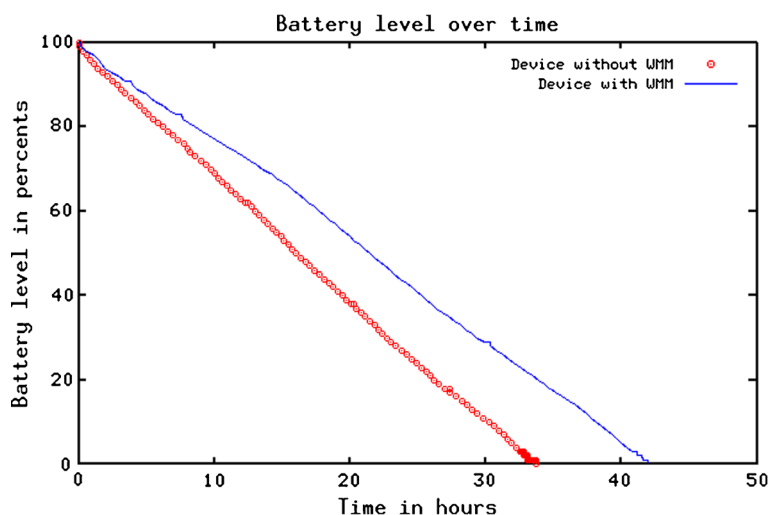


Fig. 7 Battery power saving by Working Memory Middleware

11.2 Working memory middleware and sensor service

The learning middleware approach described in [7, 33] was used to preserve the energy efficiency requirement. The Aware framework⁵ in version 2 was used as an implementation of a *sensor service*. The contextual information about threats were fetched from STM via Java API developed on the purpose of this evaluation scenario⁶.

The sensor usage model from [7] was used for the working memory middleware as the experiment was conducted by the same person for whom the model was build. We compared two prototype implementations of the mobile STM framework. One with working memory middleware implemented and the other one without. Each implementation worked on the separate (identical) device, carried by the same person.

The results shown that in a case with working memory middleware implemented, the energy saving was at the level of 20% comparing to the case without the working memory middleware. The energy saving is less impressive than described in our preliminary approach [7], because the vast amount of energy has been consumed by an inference engine, not by the sensors itself. This, again was caused by the not optimal tuProlog interpreter implementation. The results are presented in Fig. 7.

12 Summary and future work

In this paper, we presented a mobile context-aware framework that uses data gathered by mobile device sensors and perform on-line reasoning about possible threats, based on the information provided by the Social Threat Monitor.

⁵See <http://awareframework.com>

⁶See <https://bitbucket.org/mslazynski/stmjava>

The framework is designed in a service oriented architecture that includes:

1. *Inference service*, that uses HeaRT inference engine to provide on-line efficient reasoning, preserving the privacy and intelligibility requirement;
2. *Working memory middleware*, that works as knowledge cache minimizing the number of required requests to the sensors service, improving power efficiency of the entire system, preserving energy efficiency requirement;
3. *Sensor service*, that is responsible for gathering and initial preprocessing of the raw sensor data.

Major issues were encountered with fulfilling requirements defined for the inference service (see Section 9.1). Responsiveness and resource limitation that are the key requirements for the mobile device reasoning engines are still investigated. In the current approach we used tuProlog implementation which appeared to be not optimized for mobile platforms, and hence the aforementioned requirements do not hold. We investigate other implementation of Prolog interpreters, to define their weaknesses and propose solutions to solve them. The robustness of the reasoning mechanisms is also under investigation. We test approaches that use Bayesian networks and certainty factors supported by incremental data mining algorithms to allow system work in situations when limited contextual information is available.

The framework presented in this paper may be extended for additional functionalities, including:

Automatic threat reporting – The application could report anomalies in user behavior to the Social Threat Monitor system. When similar anomaly will be reported by many users, an alert will be raised. For instance, if a number of users start running from a building, there is probably a fire.

Mediation – We observe paradigm shift in terms of users of context-aware systems, where the user is no longer just a client of the "black box" system, but rather a conscious operator. This moves his role from data receiver also to the data provider, but at the same time requires him to fully understand how the system works and allow to modify and control its behavior. Methods that allow this are called mediation. We argue that together with intelligibility it is critical for providing high usability of the system, and hence we plan to incorporate this features into the STM framework.

Automatic model optimization – HeaRT inference engine provides a verification plug-in that allows detecting anomalies such as: rules subsumption, redundancy and contradiction. Hence, a mechanism that will perform automatic optimization of an existing XTT2 model could be implemented [29].

Acknowledgments Thanks to the use of mobile devices, the existing STM architecture could be extended into a fully collaborative environment as it was considered in [2]. This could be in fact combined with a collaborative system for users [31]. Moreover, the security of the device and network communication could also be formally modelled with rules [35].

References

1. Adrian WT, Bobek S, Nalepa GJ, Kaczor K, Kluza K (2011) How to reason by HeaRT in a semantic knowledge-based wiki, Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2011. USA, Boca Raton, Florida, pp 438–441. http://ieeexplore.ieee.org/xpls/abs_all.jsp?ar

2. Adrian WT, Ciekowski P, Kaczor K, Ligeza A, Nalepa GJ (2012) Web-based knowledge acquisition and management system supporting collaboration for improving safety in urban environment. In: Dziech A, Czyzewski A (eds) Multimedia Communications, Services and Security: 5th International Conference, MCSS 2012: Krakow, Poland. May 31–June 1, 2012. Proceedings, Communications in Computer and Information Science, vol. 287, pp. 1–12. [http://link.springer.com/book/10.1007/978-3-642-30721-8/page/](http://link.springer.com/book/10.1007/978-3-642-30721-8/page/1)
3. Adrian WT, Waliszko J, Ligeza A, Nalepa GJ, Kaczor K (2012) Description logic reasoning in an ontology-based system for citizen safety in urban environment. E.M.L. [et al.] (ed.) AI4KM 2012 : 1st international workshop on Artificial Intelligence for Knowledge Management at the biennial European Conference on Artificial Intelligence (ECAI 2012): August 28, 2012, Montpellier, France, pp. 63–67
4. Atzmueller M, Hilgenberg K (2013) Towards capturing social interactions with scdf: An extensible framework for mobile sensing and ubiquitous data collection Proc. 4th International Workshop on Modeling Social Media. ACM Press
5. Bardram JE (2005) The java context awareness framework (JCAF) – a service infrastructure and programming framework for context-aware applications. In: Gellersen HW, Want R, Schmidt A (eds) Pervasive Computing, Lecture Notes in Computer Science, vol 3468. Springer Berlin Heidelberg, pp 98–115. doi:[10.1007/11428572_7](https://doi.org/10.1007/11428572_7)
6. Bobek S, Nalepa GJ, Adrian WT (2013) Mobile context-based framework for monitoring threats in urban environment. In: Dziech A, Czyzewski A (eds) Multimedia Communications, Services and Security: 6th International Conference, MCSS 2013: Krakow, Poland. June 6–7, 2013. Proceedings, Communications in Computer and Information Science, vol. 368. Springer Berlin Heidelberg
7. Bobek S, Porzycki K, Nalepa GJ (2013) Learning sensors usage patterns in mobile context-aware systems Proceedings of the FedCSIS 2013 conference, Krakow, pp. 993–998. IEEE
8. Bui HH, Venkatesh S, West GAW (2001) Tracking and surveillance in wide-area spatial environments using the abstract hidden markov model. IJPRAI 15(1):177–195. <http://dblp.uni-trier.de/db/journals/ijprai/ijprai15.html#BuiVW01>
9. Chen H, Finin TW, Joshi A (2004) Semantic web in the context broker architecture PerCom, pp 277–286
10. Chen H, Perich F, Finin TW, Joshi A (2004) Soupa: Standard ontology for ubiquitous and pervasive applications. 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), Networking and Services, 22–25 August 2004, Cambridge, MA, USA, pp 258–267. <http://csdl.computer.org/comp/proceedings/mobiquitous/2004/2208/00/22080258abs.htm>
11. Chronis I, Madan A, Pentland AS (2009) Socialcircuits: the art of using mobile phones for modeling personal interactions. In: Proceedings of the ICMI-MLMI '09 Workshop on Multimodal Sensor-Based Systems and Mobile Phones for Social Computing ICMI-MLMI '09, New York, pp 1:1–1:4. doi:[10.1145/1641389.1641390](https://doi.org/10.1145/1641389.1641390).
12. Dey AK (2000) Providing architectural support for building context-aware applications. Ph.D. thesis, Atlanta. AAI9994400
13. Dey AK (2001) Understanding and using context. Personal Ubiquitous Comput 5(1):4–7
14. Dey AK (2009) Modeling and intelligibility in ambient environments. J. Ambient Intell. Smart Environ 1(1):57–62
15. Dey AK, Mankoff J (2005) Designing mediation for context-aware applications. ACM Trans Comput Hum Interact 12(1):057237–1057241. doi:[10.1145/1057237.1057241](https://doi.org/10.1145/1057237.1057241)
16. Eagle N (2006) (Sandy) Pentland, A.: Reality mining: sensing complex social systems. Personal Ubiquitous Comput 10(4):255–268. doi:[10.1007/s00779-005-0046-3](https://doi.org/10.1007/s00779-005-0046-3)
17. Ferreira D (2013) AWARE: A mobile context instrumentation middleware to collaboratively understand human behavior. Ph.D. thesis
18. Gu T, Pung HK, Zhang DQ, Wang XH (2004) A middleware for building context-aware mobile services In Proceedings of IEEE Vehicular Technology Conference (VTC)
19. Henriksen K, Indulska J, Rakotonirainy A (2002) Modeling context information in pervasive computing systems Proceedings of the First International Conference on Pervasive Computing, Pervasive '02. Springer-Verlag, London, UK, pp 646867–706693. <http://dl.acm.org/citation.cfm?id=646867.706693>
20. Hu H (2011) of Hong Kong, U.: ContextTorrent: A Context Provisioning Framework for Pervasive Applications. University of Hong Kong
21. Jaroucheh Z, Liu X, Smith S (2011) Recognize contextual situation in pervasive environments using process mining techniques. J. Ambient Intelligence and Humanized Computing 2(1):53–69
22. Kluza K, Maslanka T, Nalepa GJ, Ligeza A, Warnier M, Badica C (2011) Proposal of representing BPMN diagrams with XTT2-based business rules. In: Brazier FM, Nieuwenhuis K, Pavlin G (eds) Intelligent Distributed Computing V. Proceedings of the 5th International Symposium on Intelligent

- Distributed Computing – IDC 2011, Delft, the Netherlands – October 2011, Studies in Computational Intelligence, vol. 382, pp. 243–248. Springer-Verlag. <http://www.springerlink.com/content/d44n334p05772263/>
23. Ligeza A, Adrian WT, Ciekowski P (2012). In: Canadas J, Nalepa GJ, Baumeister J (eds) 8th Workshop on Knowledge Engineering and Software Engineering (KESE2012) at the at the biennial European Conference on Artificial Intelligence (ECAI 2012): August 28, 2012. Montpellier, France, pp 58–61. <http://ceur-ws.org/Vol-949/>
 24. Ligeza A, Ernst S, Nowaczyk S, Nalepa GJ, Szpyrka M, Furmanska WT, Czapko M, Grzesiak P (2010) Kauza, M., Krzych, M.: Towards enregistration of threats in urban environments: practical consideration for a gis-enabled web knowledge acquisition system. In: Danda J, Derkacz J, Gowacz A (eds) MCSS 2010: Multimedia Communications, Services and Security: IEEE International Conference: Krakow, 6-7 May 2010, pp. 152–158
 25. Ligeza A, Nalepa GJ (2011) A study of methodological issues in design and development of rule-based systems: proposal of a new approach. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(2):117–137. doi:[10.1002/widm.11](https://doi.org/10.1002/widm.11)
 26. Logan B, Healey J, Philipose M, Tapia EM, Intille S (2007) A long-term evaluation of sensing modalities for activity recognition Proceedings of the 9th international conference on Ubiquitous computing, UbiComp '07. Springer-Verlag, Berlin, Heidelberg, pp 771592–1771620. <http://dl.acm.org/citation.cfm?id=1>.
 27. Loke SW (2004) Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective. Knowl Eng Rev 19(3):213–233. doi:[10.1017/S0269888905000263](https://doi.org/10.1017/S0269888905000263)
 28. Nalepa G, Bobek S, Ligeza A, Kaczor K (2011) Algorithms for rule inference in modularized rule bases. In: Bassiliades N, Governatori G, Paschke A (eds) Rule-Based Reasoning, Programming, and Applications, Lecture Notes in Computer Science, vol. 6826, pp. 305–312. Springer Berlin / Heidelberg
 29. Nalepa G, Bobek S, Ligeza A, Kaczor K (2011) HalVA – rule analysis framework for XTT2 rules. In: Bassiliades N, Governatori G, Paschke A (eds) Rule-Based Reasoning, Programming, and Applications, Lecture Notes in Computer Science, vol. 6826, pp. 337–344. Springer Berlin / Heidelberg. <http://www.springerlink.com/content/c276374nh9682jm6/>
 30. Nalepa GJ (2007) Proposal of business process and rules modeling with the XTT method. In: Negru V. et al. (ed) Symbolic and numeric algorithms for scientific computing, 2007. SYNASC Ninth international symposium. September 26–29. IEEE Computer Society, IEEE, CPS Conference Publishing Service, Los Alamitos, California, Washington, Tokyo, pp 500–506
 31. Nalepa GJ (2009) PIWiki – a generic semantic wiki architecture. In: Nguyen NT, Kowalczyk R, Chen SM (eds) Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, First International Conference, ICCCI 2009, Wroclaw, Poland, October 5-7, 2009. Proceedings, Lecture Notes in Computer Science, vol. 5796, pp. 345–356. Springer
 32. Nalepa GJ (2010). In: Rutkowski Letal. (ed) Artificial Intelligence and Soft Computing: 10th International Conference, ICAISC 2010: Zakopane, Poland, June 13–17, 2010, Pt. II, Lecture Notes in Artificial Intelligence, vol. 6114, pp. 598–605. Springer
 33. Nalepa GJ, Bobek S (2014) Rule-based solution for context-aware reasoning on mobile devices. Comput Sci Inf Syst 11(1):171–193
 34. Nalepa GJ, Kluza K (2012) UML representation for rule-based application models with XTT2-based business rules. Int J Softw Eng Knowl Eng (IJSEKE) 22(4):485–524. <http://www.worldscientific.com/doi/abs/10.1142/S021819401250012X>
 35. Nalepa GJ, Ligeza A (2003) Designing reliable Web security systems using rule-based systems approach. In: Menasalvas E, Segovia J., Szczepaniak PS (eds) Advances in Web Intelligence. First International Atlantic Web Intelligence Conference AWIC 2003, Madrid, Spain, May 5-6, 2003, Lecture Notes in Artificial Intelligence, vol. 2663. Springer-Verlag, Berlin, Heidelberg, pp 124–133
 36. Nalepa GJ, Ligeza A (2005) Software engineering: evolution and emerging technologies, Frontiers in Artificial Intelligence and Applications, vol. 130, chap. Conceptual modelling and automated implementation of rule-based systems, pp. 330–340. IOS Press, Amsterdam
 37. Nalepa GJ, Ligeza A, Kaczor K (2011) Formalization and modeling of rules using the XTT2 method. Int J Artif Intell Tools 20(6):1107–1125
 38. Olguin D, Waber BN, Kim T, Mohan A, Ara K, Pentland A (2009) Sensible organizations: Technology and methodology for automatically measuring organizational behavior. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B: CYBERNETICS:43–55
 39. Ranganathan A, Campbell RH (2003) An infrastructure for context-awareness based on first order logic. Personal Ubiquitous Comput 7(6):353–364. doi:[10.1007/s00779-003-0251-x](https://doi.org/10.1007/s00779-003-0251-x)
 40. Ranganathan A, McGrath RE, Campbell RH, Mickunas MD (2003) Use of ontologies in a pervasive computing environment. Knowl Eng Rev 18(3):209–220

41. Waliszko J, Adrian WT, Ligeza A (2011) Traffic danger ontology for citizen safety web system. In: Dziech A, Czyzewski A (eds) Multimedia Communications, Services and Security, Communications in Computer and Information Science, vol. 149, pp. 165–173. Springer Berlin Heidelberg
42. Wang X, Zhang D, Gu T, Pung HK (2004) Ontology based context modeling and reasoning using owl, PerCom Workshops, pp. 18–22
43. Van Wissen B, Palmer N, Kemp R, Kielmann T, Bal H (2010) ContextDroid: an expression-based context framework for Android, Proceedings of PhoneSense 2010. <http://sensorlab.cs.dartmouth.edu/phonesense/papers/Wissen-Co>
44. Woensel WV, Casteleyn S, Troyer OD (2009) A Framework for Decentralized, Context-Aware Mobile Applications Using Semantic Web Technology
45. Ye J, Coyle L, Dobson SA, Nixon P (2008) Representing and manipulating situation hierarchies using situation lattices. *Revue d Intelligence Artificielle* 22(5):647–667. <http://ria.revuesonline.com/article.jsp?articleId=12363>
46. Ye J, Dobson S (2010) Exploring semantics in activity recognition using context lattices. *J Ambient Intell Smart Environ* 2(4):389–407



Szymon Bobek MSc (szymon.bobek@agh.edu.pl, [www](http://www.szymonbobek.com)) holds a position of a research assistant at the AGH UST in Krakow, Poland, Department of Applied Computer Science. After graduation from the university in 2009 he has continued his education and research as a member of the GEIST team. He participated in several international research projects where he was involved in tasks concerning knowledge engineering and rule based inference. His research interests are: machine learning, ambient intelligence, context awareness and knowledge engineering.



Grzegorz J. Nalepa PhD (gjn@agh.edu.pl, <http://home.agh.edu.pl/gjn>) holds a position of assistant professor in AGH University of Science and Technology in Krakow, Poland. He has been an active researcher since 1999. His work is focused on design methods for intelligent systems, as well as knowledge representation and reasoning techniques. He formulated a new design approach for intelligent rule-based systems called XTT (eXtended Tabular Trees), as well as the Semantic Knowledge Engineering design and analysis methodology for knowledge-based systems. He authored and co-authored over 150 publications (see here) from the domains of rules and knowledge engineering, intelligent systems, semantic web, business intelligence and software engineering. His fields of interest also include computer security and operating systems. Moreover he is interested in cognitive aspects of knowledge representation and reasoning in intelligent systems within the scope of cognitive science. He gave invited lectures in the University of Almeria, Spain in 2011 and 2012. He presented an invited tutorial on FedCSIS 2012 entitled Semantic Knowledge Engineering for Business Intelligence: concepts and tools. In 2009 along with professor Antoni Ligza he created the GEIST research team, which he has been leading since then. He has been actively involved in number of research projects, including Regulus, Mirella, Adder, HeKatE and recently INDECT, BIMLOQ, Parnas and Prosecco. He coordinated several projects including HeKatE, BIMLOQ, Parnas and Prosecco. From 2012 his work is focused on two main application areas: business intelligence, including business rules and process modeling and analysis, and ambient intelligence including context aware applications for mobile devices. He has been involved in over 30 international conferences and workshops. Since 2008 he co-organizes the Knowledge and Software Engineering Workshop (KESE) at KI, the German AI conference, and more recently on the Spanish CAEPIA, as well as ECAI 2012. He is the President of the Polish Artificial Intelligence Society (PSSI <http://pssi.agh.edu.pl>), member of ECAI. Dr Nalepa holds PhD and Dsc (habilitation) from the AGH UST. He has been lecturing computer science courses in number of polish universities. He has been working with commercial companies preparing professional trainings in computer security and operating systems. He also took part in a number of curricula preparations, including graduate and postgraduate studies, for several universities. From 2011 he has been a coordinator for several Erasmus exchange agreements. He is an active user and supporter of free software and open source software. For a full scientific resume see <http://home.agh.edu.pl/~gjn/wiki/en:research>.



Antoni Ligeza a full professor in the domain of computer science at the AGH University of Science and Technology at Krakow, Poland. His principal area of investigation is Artificial Intelligence and Knowledge Engineering. He lectures on knowledge engineering, databases, Prolog, automated diagnosis, discrete mathematics and logics. He is a member of ACM and IEEE Computer Society. He is author and co-author of over 200 research publications, including international conferences, journals, chapters in books. His recent book *Logical Foundations for Rule-Based Systems* was issued by Springer in 2006, it covers issues ranging from logical bases, propositional, attributive and first-order logics, through various forms of rule-based systems to design and verification issues. It presents a novel approach based on XTT (eXtended Tabular Trees) and ARD (Attribute Relationship Diagrams) for efficient design and implementation of complex rule-based systems. The XTT approach can be applied to develop the control system for autonomous robots. It offers features like hierarchical control, rule-based control, context-switching, backtracking, internal state representation, and many other. The control algorithm can be verified for completeness and consistency; it can be also easily extended and modified. Antoni Ligeza actively participated in numerous national and international projects (KBN, TEMPUS, POLONIUM); he was the head of KBN-Regulus Project2, the Mirella Project on XTT (eXtended Tabular Systems) and recently HeKatE on hybrid knowledge engineering, and INDECT. He was visiting professor and he worked in Denmark (Technical University of Lyngby) 4 months, in France (LAAS of CNRS, Toulouse; University of Nancy I, Nancy; CRIL Lens; University of Caen, Caen) for about two years in total and in Spain (University of Balearic Islands, Palma de Mallorca; University of Girona, Girona) for about one year. Professor Antoni Ligeza developed theory for reverse plan generation (backward planning) (Artificial Intelligence, 1990), the backward dual resolution method for automated theorem proving and completeness verification of rule-based systems (IJCAI'93), methodology for verification and design of rule-based systems based on the so-called psi-trees (ECAI Workshop 1996), concepts of granular sets and granular relations (AIMeth'02, IIS (Internet Information Services)'02, ANNIE'03), attributive granular logic (EMCSR'06) and theory for tabular rule-based systems (Springer, 2006). He also worked in areas such as Operational Research, Control Theory, Decision Theory, Diagnostics, Databases, and many other. He supervised numerous Ph.D. theses. He also serves as a reviewer for numerous international conferences and journals.



Weronika T. Adrian MSc (wta@agh.edu.pl, www) is a research assistant at AGH University of Science and Technology. She is a member of the GEIST Research Group, a founder member of the Polish Artificial Intelligence Society and a member of IEEE. She has cooperated with GEIST members since 2008. Her main research interests cover Semantic Web technologies and logic programming. She is an author and co-author of 28 publications (the complete list is available in the AGH bibliographic system). Selected publications can be browsed in the Researcher ID and DBLP systems. She has been involved in several national and international research projects: HeKatE, BIMLOQ, INDECT, Prosecco; and EU programs: Knowledge and Practice, it2edu, SPiN. She is a laureate of the first edition of Top 500 Innovators project sponsored by Polish Ministry of Science and Higher Education within which she completed a program on Science-Management-Commercialization at Stanford University (for more information please see a testimonial at SCPD Website or read the blog).



Krzysztof Kaczor holds a position of a research assistant at the AGH UST in Krakow, Poland, Department of Automatics. He has received his MSc degree from AGH UST in 2008. In his master thesis he focused on design and implementation of a unified rule base editor allowing for visual modeling of the XTT2 rule bases. Since 2007 he has been actively involved in the number of research projects, where his work was mainly related with the visual and formal knowledge representation and processing (Hekate, Rebit, Bimloq). Currently, his work and PhD dissertation concern the development of unified and formalized model for rule representation allowing for semantic business rule interoperability. For the list of his publications see: the bpp AGH system, the Google Scholar, the DBLP Bibliography or my ResearcherID profile (based on the Web of Knowledge).